Implementing Lane Margining in a Heterogeneous System

Trupti Gowda

Sr. Field Application Engineer

PLDA

Disclaimer

<u>Presentation Disclaimer</u>: All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG®.

Summary

- **Introduction**
- **RX Margining Basics Reminder**
- **MAC-PHY cooperation**
- **Challenges**
- **Conclusion**

Introduction

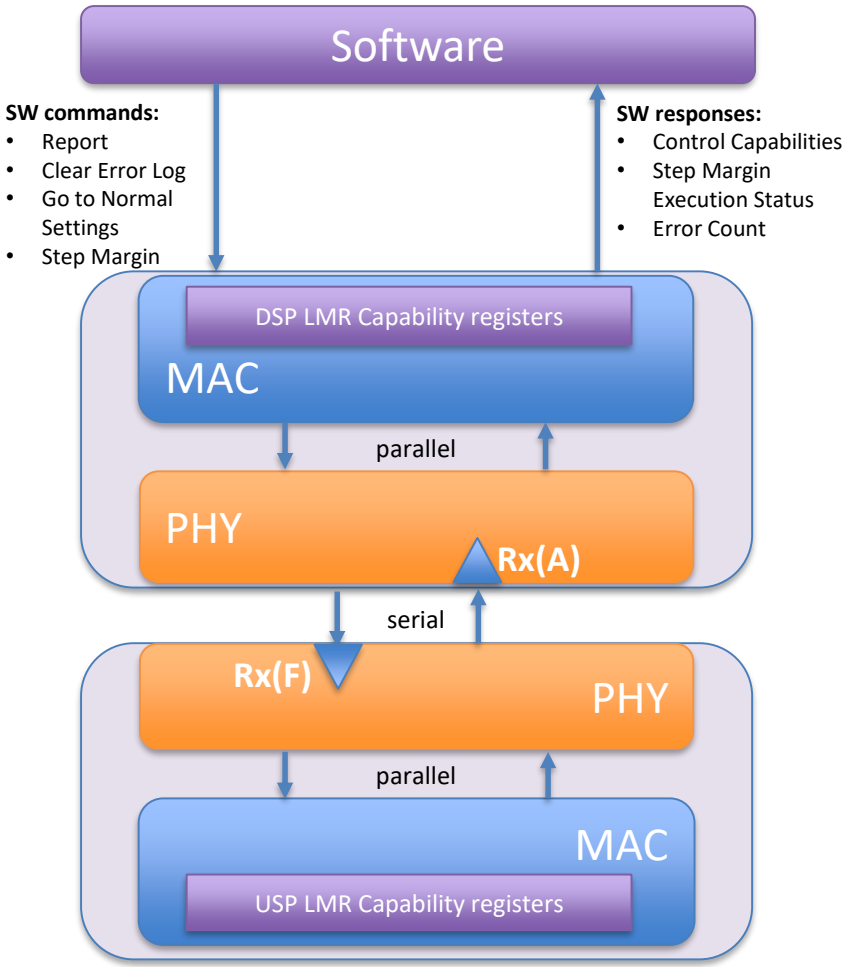
- **Lane margining at Receiver is a new feature that has been introduced in PCIe® 4.0 release**
  - Like every new feature, it raises questions and causes problems at its early implementation
  - It is defined at the Physical Layer and its functionality is spread across both the PCIe controller and the PHY, thus requiring a handshake mechanism which could be a source of errors

Introduction

- **To ensure interoperability one must account for the various challenges that could be faced during the implementation of this feature such as:**
  - No strict definition of MAC vs PHY responsibilities
  - Different interpretation of the standard
  - Lack of PHY I/F definition in the early stage of the protocol release
  - Different set of handshake commands supported on both ends
  - …

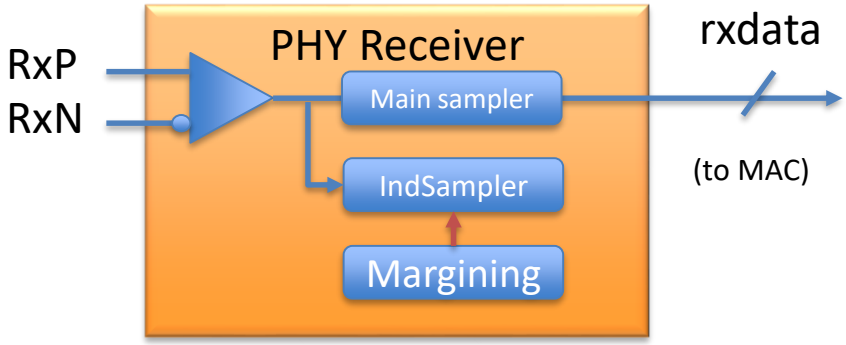
RX Margining Basics Reminder

- **New test mode with no external hardware**
- **Software controlled**
- **Run in L0 @ 16GT/s rate**
- **Counting either bit or logical errors**
- **Using the main data sampler or an independent one**

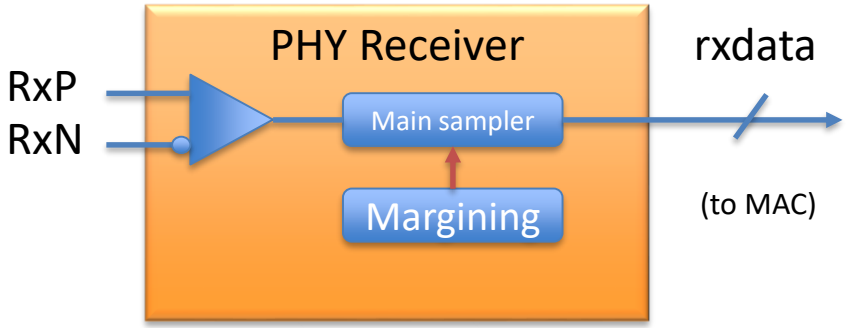


Margining Methods

- **Margining with independent sampler**



- **Margining with actual data sampler**



Margining Methods Main Differences

Independent Sampler	Main Sampler
Bit errors counted	Logical errors counted
Not producing errors in data stream	Producing errors in data stream
Measurement done by PHY	Measurement done by MAC
Exceeding the error limit *optionally* ends margining	Exceeding the error limit *mandatory* ends margining

Did you consider…?

- **PCIe specification does not provide strict definition of MAC vs PHY responsibilities**
  - ⇒ Different interpretations are possible in the cases of:
    - Sample count maintenance
    - Error count reset following “Go to Normal Settings”
    - Error count reset upon step change

MAC maintaining a Sample count?

- **Is the MAC required to maintain and update the Sample count if not done by the PHY?**

  -> No. This capability is advertised in each PHY datasheet and only one of the two Sample reporting methods is allowed within a device:

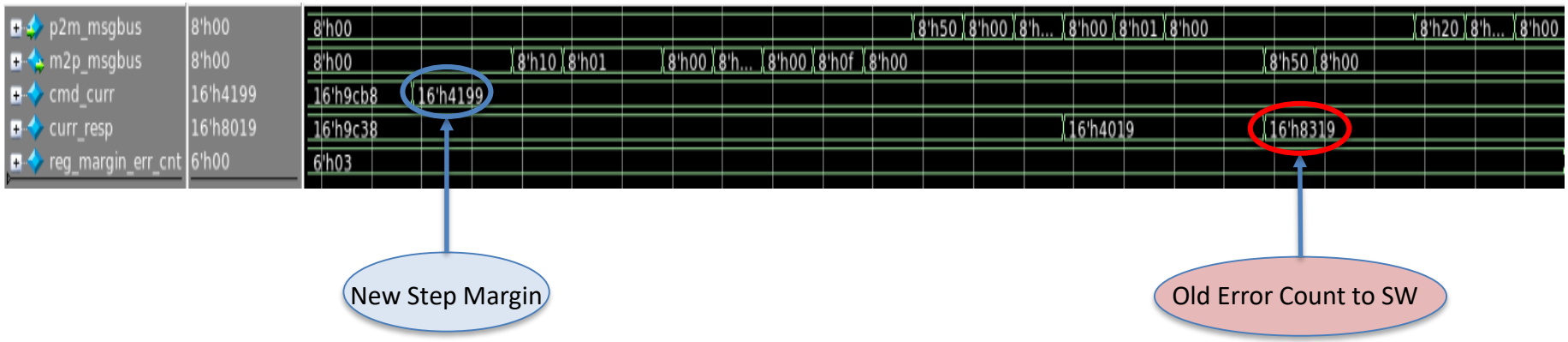
  - Sample count
  - Sampling rates

Who clears the bit errors count when margining is stopped?

- **Scenario:**
  - MAC receives from Software a Clear Error Log command as part of the stop margining sequence
  - MAC instructs the PHY to reset the accumulated number of bit errors
  - MAC expects to receive 0 value count update from PHY

Who clears the bit errors count when margining is stopped?

- **Impact:**
  - MAC reporting to SW incorrect score for the next tested step/direction due to keeping the old value



Who clears the bit errors count when margining is stopped?

- **What could be done?**
  - Error count could be cleared internally in MAC without expecting the receipt of the reset value

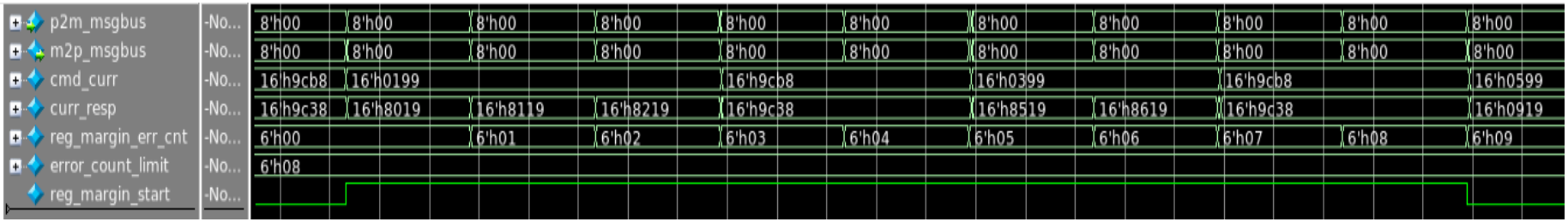
Error Count Reset for a New Step

- **Standard does not explicitly define the case of autonomous error count reset when margining is running and a new step is received**

- **Scenario:**
  - Step command received
  - Margining started
  - Errors recorded
  - New Step command received with only the offset changed, so margining not stopped
  - Number of recorded errors increased from previous value

Error Count Reset for a New Step

- **Impact:**
  - Error count accumulated for different margin steps
  - Error limit reached and margining ended prematurely



Error Count Reset Before a New Step

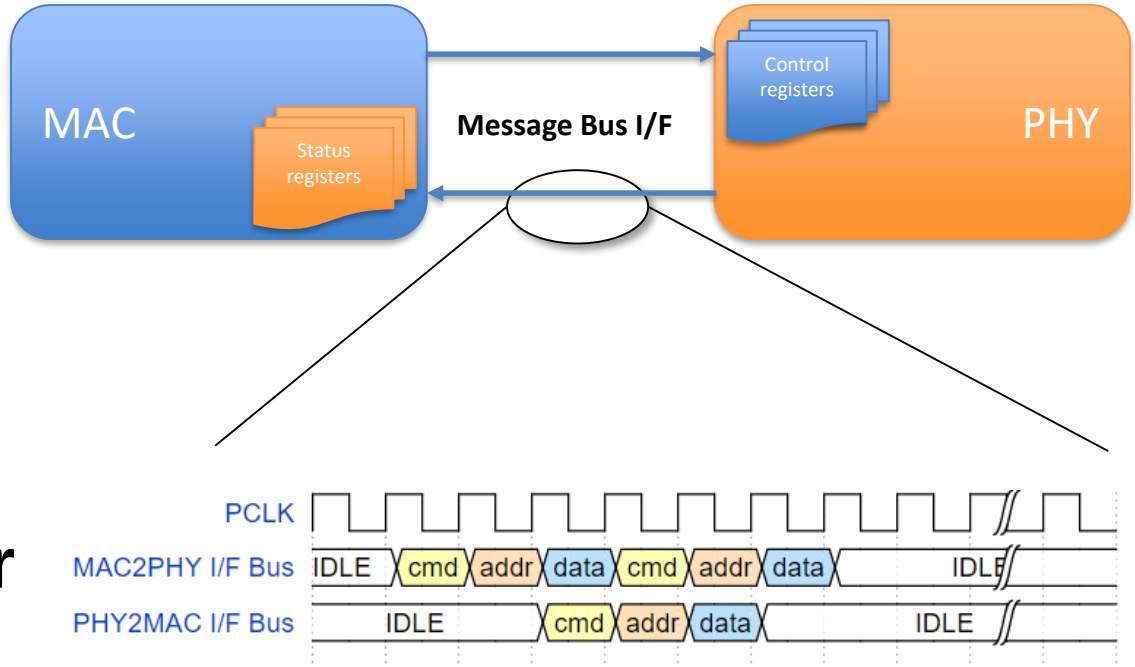
- **Solution:**
  - MAC should reset the error count (and sample count) on every new Margin Step command
    - This ensures that the provided measurement is for a single sample position

MAC-PHY Cooperation

- **Majority of vendors base their implementation on a message bus interface:**
  - MAC communicating translated SW commands to targeted PHY Receiver
  - PHY returning status of commands execution

MAC-PHY Cooperation

- **New PHY I/F signals**
- **MAC & PHY registers**
- **Address-based transactions for commands and responses**



MAC to PHY Instructions Issues

- **MAC-PHY handshaking is done with the use of a set of message bus interface transactions with no error recovery**
  - MAC does not retransmit a command if no response received from PHY

MAC to PHY Instructions Issues

- **Possible scenarios include:**
  - Unsupported transaction type
  - Different set of PHY level register fields
  - Margin offset change before Margin Status update
  - Margin Status update delay
  - Autonomous Margin stop upon rate change

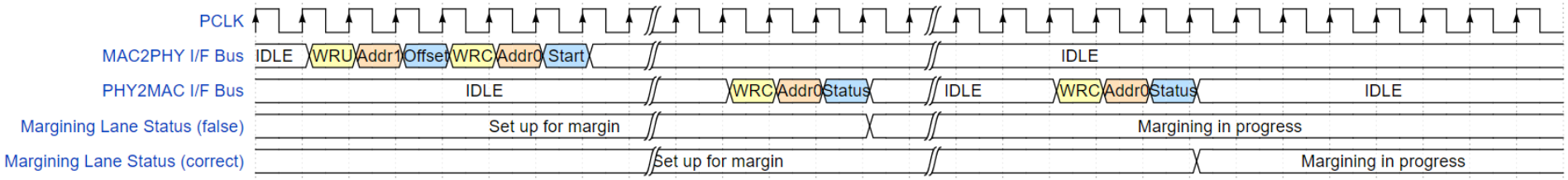
MAC to PHY Instructions Issues Transaction Types

- **What if one side does not support a transaction type used on the other?**

Supported set in MAC	Supported set in PHY
ACK	ACK
WRC	WRC
WRU	

MAC to PHY Instructions Issues Transaction Types

- **Scenario:**
  - MAC issuing WRU (offset) + WRC (start)
  - PHY returning a response for the first command
  - MAC interpreting it as a response to the second one
  - MAC updating Step Margin Execution Status from “Setup” to “Margining in Progress” earlier than it actually becoming true



MAC to PHY Instructions Issues Transaction Types

- **Impact:**
  - If SW relies on the “Margining in progress” step margin execution status when determining the sample size in the case when the Sample Count is not being provided by the PHY, BER could be slightly more optimistic, although negligibly

MAC to PHY Instructions Issues Transaction Types

- **Solution:**
  - Implementation of additional ability to operate with reduced set of command types when level of support in peer is lower

MAC to PHY Instructions Issues Register Fields

- **What if PHY I/F register fields in PHY and MAC differ?**

Addr	Bits	New	Old
0x000	7:5	Reserved	Reserved
	4	Reserved	Sample Count Reset
	3	Sample Count Reset	Error Count Reset
	2	Error Count Reset	Margin Direction
	1	Margin Voltage/Timing	
	0	Start Margin	Start Margin
0x001	7	Margin Direction	Reserved
	6:0	Margin Offset	Margin Offset

MAC to PHY Instructions Issues Register Fields

- **Impact**
  - Commands triggering unintended behavior
  - MAC reporting back to SW incorrect measurements
  - Margining fail if at all started

MAC to PHY Instructions Issues Register Fields

- **Solutions:**
  - Same set of registers aligned in MAC and PHY
    - Limits the possible interoperability with other vendors
  - Both versions supported in the MAC
    - Much cheaper to add modifications to the controller
    - Controller is the initiator of these transactions
    - Considering support for future versions

Margin Offset Change Before Margin Status Update

- **Case:**
  - MAC sending command with new step received
  - MAC detecting framing error and directing Link to Recovery
  - PHY not yet returning confirmation for executing the received command
  - MAC resetting sampling offset back to default upon Link entering Recovery before previous change acknowledged

Margin Offset Change Before Margin Status Update

- **Impact:**
  - PHY failing to return Margin Status at all
  - Margining remaining at “Set up” longer than allowed
  - Margining fail

Margin Offset Change Before Margin Status Update

- **How to solve?**
  - MAC waiting for previous change acknowledge from PHY before changing the offset

Margin Status Update Delay

- **Case:**
  - Margining with the main sampler
  - MAC directing PHY to default sample position when entering Recovery
  - MAC returning PHY to directed step position after reaching back L0
  - Margin Status update taking longer than next error that causes another link retrain

Margin Status Update Delay

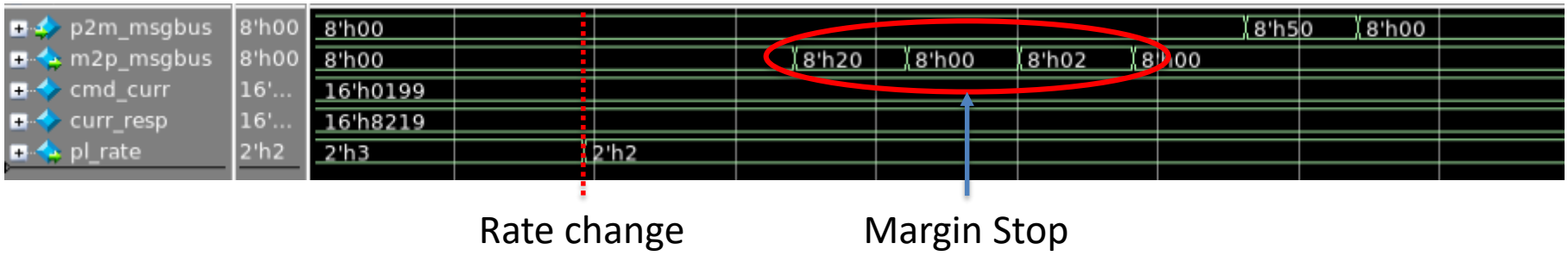
- **Where does this lead?**
  - MAC not resetting the offset when Link is in Recovery
  - PHY continuing sampling with the changed position
  - Link retrain fail due to increased number of bit errors
  - Margining fail

Margin Status Update Delay

- **How to avoid?**
  - Still working with PHY vendors to conclude on this case
  - Questions remaining:
    - Should MAC be restrained by PHY?
    - Should PHY be able to handle such requests?
    - How to synchronize both in order to obey the standard?

Autonomous Margin Stop Upon Rate Change

- **Scenario:**
  - Standard defines that margining ends when the rate is no longer 16 GT/s
  - MAC issues a command on PHY I/F to instruct the PHY to stop margining after the rate change
  - PHY fails to respond
  - Link training fails



Autonomous Margin Stop Upon Rate Change

- **How to overcome?**
  - MAC needs to ensure all instructions are issued while the Link is still operating at PCIe 4.0
  - MAC needs to wait for all response from PHY for all commands before performing the rate change

Conclusion

- Considering these key points prior to moving to the testing phase is essential for ensuring seamless integration and interoperability between a PCIe controller and a PHY

- As a result, this will permit correct measurement of the quality of the received signal in a live system, integrating these components

Thank you for attending the PCI-SIG Developers Conference 2018.

For more information, please go to www.pcisig.com

Don't forget to submit your feedback via the mobile app!

Download the **Crowd Compass** app and then search for **PCI-SIG Developers Conference** or entering the following URL into your mobile browser: https://crowd.cc/s/1rKy0
Enter event code: **DevCon2018**

Alternatively, access here: https://crowd.cc/pcisig2018

Note: Create an account within the app so Admin knows who to contact if selected as the prize winner.

Each session feedback is provided is equivalent to 1 raffle entry (up to 11 sessions).
General survey feedback = 1 raffle entry.